Original version at `http://iram-institute.org/medias/uploads/class-herschel-fits.pdf`



IRAM Memo 2015-3
Importing Herschel-FITS into **CLASS**

S. Bardeau[1], D.Teyssier[2], D.Rabois[3], J. Pety[1,4]

1. IRAM (Grenoble)
2. ESAC (Madrid)
3. IRAP (Toulouse)
4. Observatoire de Paris

October, 28st 2016
Version 1.0.1

Abstract

Herschel/HIFI data were already readable by **CLASS** after conversion into specific FITS file by the HiClass task from HIPE. These FITS files are however not the ones served by the Herschel Science Archive as result of the standard data processing. The **CLASS** community thus requested the possibility to fill Herschel/HIFI archive science products directly into **CLASS**.

In a nutshell, the new filler automatically recognizes the Herschel/HIFI FITS level 2.0 and 2.5. The standard **CLASS** sections (`General, Position, Spectroscopy, Calibration`) are filled from the FITS header parameters. A dedicated Herschel section is added to the **CLASS** Data Format in order to attach HIFI specific metadata. The data are filled as one or several **CLASS** spectra depending of the FITS level and/or context (On-The-Fly, spectral survey, etc). The associated FLAG and LINE arrays are filled as associated arrays (new **CLASS** feature documented in a separate memo) to each spectrum data. This memo describes all these points in details.

Keywords: **CLASS** Data Format
Related documents: **CLASS** documentation, **CLASS** Associated Arrays.

Contents

1 Quick start guide **3**
1.1 Supported HIPE versions . . . 3
1.2 Importing Herschel-HIFI FITS . . . 3
1.3 Observation header . . . 3
1.4 Associated Arrays . . . 4
1.5 Setting the source velocity . . . 4
1.6 Going from T_A^* to $T_{\rm mb}$ . . . 4

2 Look-up tables **5**
2.1 Intensity array . . . 5
2.2 General section . . . 6
2.3 Spectroscopic section . . . 6
2.4 Position section . . . 7
2.5 Calibration section . . . 7
2.6 Frequency Switch section . . . 8
2.7 Herschel-HIFI section . . . 8
2.8 Associated Arrays . . . 8

3 Technical details **10**
3.1 Detecting an Herschel-HIFI FITS . . . 10
3.2 FITS level . . . 10
3.3 Cards and metacards . . . 10

4 Acknowledgements **10**

1 Quick start guide

You can import Herschel-HIFI FITS files if you are using a **CLASS** version equal or newer than oct15. Older versions of **CLASS** will warn you about unsupported data and will give you hand back without importing anything.

1.1 Supported HIPE versions

The filler is able to import data created by versions[1] of the HIPE software equal or newer than 12. Older versions are rejected by **CLASS**. However, it is recommended to retrieve the latest Herschel-HIFI FITS files (version 14, expected mid-2016) from the Herschel Science Archive, as there were several fixes (*e.g.* missing keywords) added since the versions 12 and 13. This means the resulting spectra in **CLASS** will be incomplete (*e.g.* null values, in particular in the Herschel-HIFI section) or approximate for versions 12 and 13. If this happens, **CLASS** will issue warnings when it had to choose the default value for a parameter. Defaults are documented in the look-up tables in the Section 2.

1.2 Importing Herschel-HIFI FITS

Once you have installed a correct version of **CLASS**, importing Herschel-HIFI FITS is as simple as opening a **CLASS** output file and calling the command `FITS`, *e.g.*

```
LAS> file out myfile.hifi single
LAS> fits read herschel-hifi.fits
```

Depending on the observing mode and on the pipeline level used in the *Herschel Common Science System* (HCSS), **CLASS** will produce one or more spectra from a single FITS file. Once this is done, you can retrieve the spectra as usual in **CLASS**, *e.g.*:

```
LAS> file in myfile.hifi
LAS> find
LAS> list
LAS> get first
```

The first spectrum is stored in the `R` buffer and is ready for reduction with the **CLASS** analysis tools.

1.3 Observation header

The relevant metadata have been exported from the Herschel-HIFI FITS to the **CLASS** observation header, namely in:

- the General section in `R%HEAD%GEN%`,
- the Position section in `R%HEAD%POS%`,
- the Spectroscopic section in `R%HEAD%SPE%`,
- the Frequency Switch section in `R%HEAD%FSW%` if relevant,
- a subset of the Calibration section in `R%HEAD%CAL%`, and
- the new Herschel-HIFI section in `R%HEAD%HER%`, where the metadata specific to this instrument can be found.

A raw overview of the whole header can be displayed with the command `DUMP`. Each section can be listed (name and values) with the command `EXAMINE`. We present in the section 2 how these sections were filled from the Herschel-HIFI FITS headers.

[1] The version can be found in the header card `CREATOR`.

1.4 Associated Arrays

The spectra imported from Herschel-HIFI FITS are also one of the firsts to use the so-called *Associated Arrays* in the **CLASS** observation. These arrays can be found under the structure `R%ASSOC%`. **CLASS** can understand those special arrays (see below where the dedicated documentation can be found). In practice, the Herschel-HIFI spectra provides the usual intensity array (*i.e.* in the `RY` variable), but also 2 arrays of the same size (same number of channels):

- the `BLANKED` Associated Array is an integer flag array indicating if the `RY` values should be blanked out (1) or not (0), under the user choice. This array is similar the usual *bad* values in **CLASS**, except that the actual values (*i.e.* before blanking) are still available. The typical use of this array is:

```
LAS> LET RY R%HEAD%SPE%BAD /WHERE R%ASSOC%BLANKED%DATA.EQ.1
```

- the `LINE` Associated Array is an integer flag indicating if the `RY` values are in (1) or out (0) a spectral line window. If the option `/ASSOCIATED` is invoked with `SET WINDOW`, the command `BASE` will use this array (*i.e.* ignore channels in a spectral window) for baselining (see `HELP SET WINDOW` for details). Typical use is:

```
LAS> SET WINDOW /ASSOCIATED  ! Will use the LINE Associated Array
LAS> SHOW WINDOW             ! Check you will use the LINE Associated Array
LAS> DRAW WINDOW             ! Draw the LINE Associated Array on the spectrum
LAS> BASE                    ! Use the LINE Associated Array during baselining
```

Note that the *Associated Arrays* are removed by the commands that do not know how to process them. In this case, a warning is issued. This is being solved as the *Associated Arrays* are more and more integrated in **CLASS**. In order to know the current status, please check the dedicated documentation for your current **CLASS** version in the widget menu > Help > CLASS > Associated Arrays.

1.5 Setting the source velocity

The spectra are imported assuming a zero-valued velocity of the source in the LSR frame. This most probably implies that the line frequencies will appear redshifted. The standard use of **CLASS** is to deliver the frequency axis in the source frame. To stick to this use, you have to modify the source velocity for each spectrum early in the reduction process with the command `MODIFY VELOCITY` (see associated `HELP` for details). In order to help you, the source systemic velocity of the source provided in the HIFI observing setup by the user is imported into the variable `R%HEAD%HER%VINFO`. If its value is correct, the command

```
LAS> MODIFY VELOCITY R%HEAD%HER%VINFO
```

will shift the frequency axis to the rest frame of this velocity.

1.6 Going from T_A^* to $T_{\rm mb}$

In **CLASS**, the brightness scale is assumed to be expressed in T_A^* when the forward and beam efficiencies are equal. To check this, you can type

```
LAS> EXAMINE R%HEAD%CAL%BEEFF R%HEAD%CAL%FOEFF
R%HEAD%CAL%BEEFF = 0.9600000    ! Real    GLOBAL RO
R%HEAD%CAL%FOEFF = 0.9600000    ! Real    GLOBAL RO
```

In this case, if the source properties are better represented by the main temperature scale ($T_{\rm mb}$), you can convert the scale from T_A^* to $T_{\rm mb}$ with

```
LAS> EXAMINE R%HEAD%HER%ETAL R%HEAD%CAL%FOEFF  ! Check that they have identical values
R%HEAD%HER%ETAL = 0.9600000     ! Real    GLOBAL RO
R%HEAD%CAL%FOEFF = 0.9600000    ! Real    GLOBAL RO
LAS> EXAMINE R%HEAD%HER%ETAMB                ! Check that the beam efficiency is sensible
R%HEAD%HER%ETAMB =   0.6223738  ! Real    GLOBAL RO
LAS> MODIFY BEAM_EFF R%HEAD%HER%ETAMB        ! Scale conversion
I-MODIFY,  Former beam efficiency: 0.9600, new: 0.6224
```

The `MODIFY BEAM_EFF` command will multiply the spectrum by `R%HEAD%CAL%FOEFF / R%HEAD%HER%ETAMB` and change the associated header parameters accordingly, *e.g.* baseline rms and $T_{\rm sys}$.

2 Look-up tables

We describe here, section by section, how the **CLASS** observation header is filled from the Herschel-HIFI FITS header and data. You can refer to the **CLASS** or Herschel-HIFI documentations for the exact specification of each component. The concept of FITS cards and metacards is explained in the section 3.3.

Table 1: Look-up table between the **CLASS** General section and the Herschel-HIFI FITS. Depending on the FITS structure (pipeline level), some elements found in a FITS header or in a FITS binary table. FITS names are case sensitive. The elements not described here are set to their usual undefined value in **CLASS**.

Name	Type	Value	Default
`num`	I*8	Automatic	
`ver`	I*4	1	
`teles`	C*12	`HIF-LL-SP-BB` where:	
		- `LL` is `00` (level 2.5) or the subband number in the column `flux_*` name (level 2.0),	`00`
		- `S` is the spectrometer, `W` for WBS or `H` for HRS (1st letter of the card `BACKEND`, else of the metacard `polarization`)	Blank
		- `P` is the polarization, `V` or `H` (5th letter of the card `BACKEND`, else of the metacard `polarization`)	Blank
		- `BB` is the band name (2 first letters of the card `BAND`)	Blank
`dobs`	I*4	Integer part of average of card `DATE-OBS` and card `DATE-END`	Current day
`dred`	I*4	Current day	
`kind`	I*4	**CLASS** internal code for spectroscopic data	
`qual`	I*4	**CLASS** internal code for unknown quality	
`scan`	I*8	Metacard `bbtype` (level 2.5) or column `bbtype` (level 2.0)	0
`subscan`	I*4	Metacard `bbnumber` (level 2.5) or column `bbnumber` (level 2.0)	0
`ut`	R*8	Fractional part of average of card `DATE-OBS` and card `DATE-END`	Current time
`tau`	R*4	0.0	
`tsys`	R*4	Metacard `tsys_median` (level 2.5) or column `tsys_median` (level 2.0)	0 K
`time`	R*4	Metacard `integrationTime` (level 2.5) or column `integrationTime` (level 2.0)	0 sec

2.1 Intensity array

The intensity array (`RY`) is extracted from one or more columns in the binary table. There are 2 possibilities:

- the `flux_*` columns (where `*` is the subband number), each of them producing a different spectrum (level 2.0). In this case, one row of the column contains all the channel values. The associated frequency array is found in the column `lsbfrequency_*` or `usbfrequency_*`, else an error is raised.

- the `flux` column, producing a single spectrum (level 2.5). In this case one row of the column contains 1 channel value. The associated frequency array is found in the column `frequency` if available, else `wave`, else an error is raised.

All spectra (flux and frequency arrays) are always sorted in ascending frequency (if needed), before setting the spectroscopic section as described in the Table 2. The Not-a-Number (`NaN`) values (if any) in the flux array are patched to the `bad` value defined in the **CLASS** header.

2.2 General section

A subset of the general section is filled. The elements *sidereal time*, *azimuth*, and *elevation* are left to the **CLASS** internal defaults as they have no meaning for space-based observations. Note that for spectral scans (frequency surveys), the `integrationTime` is set by HIPE to the integration time of the individual tunings multiplied by the median redundancy accross the frequency axis.

2.3 Spectroscopic section

Table 2: Look-up table between the **CLASS** Spectroscopic section and the Herschel-HIFI FITS. Depending on the FITS structure (pipeline level), some elements found in a FITS header or in a FITS binary table. FITS names are case sensitive.

CLASS	Type	Value
`line`	C*12	`DECON_SSB` if Card `OBS_MODE` contains `SScan` and if Card `CLASS___` contains `Spectrum1d`, else `XXXXXXXX_YYY` where `XXXXXXXX` is the `lofreq` in GHz and `YYY` is the metacard `sideband` value (`GHZ` if not defined)
`nchan`	I*4	Card `NAXIS2` (level 2.5) or Card `TDIM*` for column `flux_*` (level 2.0)
`restf`	R*8	Value at `rchan` in column `frequency` or `wave` (level 2.5) or value at `rchan` in column `lsbfrequency_*` or `usbfrequency_*` (level 2.0)
`image`	R*8	2*`lofreq-restf` if `lofreq` is defined, **CLASS** internal code for undefined value otherwise
`doppler`	R*8	0.d0
`rchan`	R*8	ceiling(`nchan`+1)/2 (*i.e.* middle of spectrum)
`fres`	R*8	Absolute of spacing in column `frequency` or `wave` (level 2.5) or absolute of spacing in column `lsbfrequency_*` or `usbfrequency_*` (level 2.0)
`vres`	R*8	$-c\times$ `fres` / `restf`
`voff`	R*8	0.d0
`bad`	R*4	-1000.0
`vtype`	I*4	code for LSR referential if Metacard `freqFrame` is `LSRk`, code for unknown referential if Metacard `freqFrame` is `source`, error otherwise.
Not in header:		
`lofreq`	R*8	Metacard `LoFrequency` else `LoFrequency_measured` (level 2.5) or Column `LoFrequency` else `LoFrequency_measured` (level 2.0), 0.d0 otherwise.

The exact status of the spectroscopic axes after HIFI calibration is available here:
`http://herschel.esac.esa.int/twiki/pub/Public/HifiCalibrationWeb/freq_vel_1.1.pdf`

In the context of **CLASS** we can note that the frequencies found in the Herschel-HIFI FITS are expressed in the LSR frame[2] (hence the null Doppler factor). The source velocity is assumed to be 0 at

[2]LSR frame, except for the Solar System Objects for which Herschel-HIFI FITS uses the source frame ($v = 0$); in this

import time; this is consistent with the null Doppler factor. It is the responsibility of the user to apply the needed `MODIFY VELOCITY` correction to apply the frequency shift implied by the source systemic velocity (see section 1.5), *i.e.*, to get the frequency axis in the source frame.

Moreover, at import time, the default rest frequency of the spectrum is arbitrarily set to the middle of the spectrum bandpass. Here again, it is the responsibility of the user to apply the needed `MODIFY FREQUENCY` correction to center the velocity axis around the line rest frequency under consideration.

2.4 Position section

The position section is described in the Table 3. Note that the source position is described as the offsets between the actual (reconstructed) position (`RA` or `longitude`, `DEC` or `latitude`) and the nominal (commanded) position (`RA_NOM`, `DEC_NOM`). This also means that for Solar System Objects (SSO), the offsets are not expressed in the comoving frame, but from a fixed position on sky.

Table 3: Look-up table between the **CLASS** Position section and the Herschel-HIFI FITS. Depending on the FITS structure (pipeline level), some elements found in a FITS header or in a FITS binary table. FITS names are case sensitive.

CLASS	Type	Value	Default
`sourc`	C*12	Card `OBJECT`	`UNKNOWN`
`system`	I*4	**CLASS** internal code for Equatorial system	
`equinox`	R*4	Card `EQUINOX`	Error
`proj`	I*4	**CLASS** internal code for Radio projection	
`lam`	R*8	Card `RA_NOM`	Error
`bet`	R*8	Card `DEC_NOM`	Error
`projang`	R*8	0.d0	
`lamof`	R*4	Offset in current projection between card `RA_NOM` and card `RA` (level 2.5) or column `longitude` (level 2.0)	0, *i.e.* actual position defaults to nominal
`betof`	R*4	Offset in current projection between card `DEC_NOM` and card `DEC` (level 2.5) or column `latitude` (level 2.0)	0, *i.e.* actual position defaults to nominal

2.5 Calibration section

The **CLASS** calibration section is activated and partially filled if the metacard `temperatureScale` is defined. The rules are detailed in Table 4.

Table 4: Look-up table between the **CLASS** Calibration section and the Herschel-HIFI FITS. Depending on the FITS structure (pipeline level), some elements found in a FITS header or in a FITS binary table. FITS names are case sensitive. The elements not described here are set to their usual undefined value in **CLASS**.

CLASS	Type	Value	Default
`beeff`	R*4	Metacard `forwardEff` if metacard `temperatureScale` is `T_A*`, else metacard `beamEff` if metacard `temperatureScale` is `T_MB`, else error	0.0
`foeff`	R*4	Metacard `forwardEff`	0.0
`gaini`	R*4	Metacard `*sbGain` (`usb` or `lsb`)	0.0
`tchop`	R*4	Metacard `hbbTemp` (level 2.5) or 1st value in Column `hot_cold` (level 2.0)	0.0
`tcold`	R*4	Metacard `cbbTemp` (level 2.5) or 2nd value in Column `hot_cold` (level 2.0)	0.0

case, the velocity type is set as *unknown* in **CLASS**.

2.6 Frequency Switch section

This section is enabled and filled in the **CLASS** observation header if the FITS card `OBS_MODE` contains the string `FSwitch` and if a non-null LO throw value is found in the FITS data.

Table 5: Look-up table between the **CLASS** Frequency Switch section and the Herschel-HIFI FITS. Depending on the FITS structure (pipeline level), some elements found in a FITS header or in a FITS binary table. FITS names are case sensitive. Some values are not read from the FITS but explicitly set.

CLASS	Type	Value	Default
`nphas`	I*4	2	
`swmod`	I*4	Metacard `isFolded` (T/F), translated to **CLASS** internal code	code for False
`decal[1]`	R*8	0.d0	
`duree[1]`	R*4	1.0	
`poids[1]`	R*4	+0.5	
`decal[2]`	R*8	Column `LoThrow` else `lothrow` (level 2.0), or Metacard `LoThrow` else `loThrow` (level 2.5)	
`duree[2]`	R*4	1.0	
`poids[2]`	R*4	-0.5	

2.7 Herschel-HIFI section

The Herschel-HIFI section is filled from the FITS according to the table 7. Some of the parameters here are available for bookkeeping purpose, some others are essential for data analysis. In particular, the sideband gain coefficients[3] will be used for double-side band deconvolution.

2.8 Associated Arrays

CLASS imports each Herschel-HIFI spectrum together with the 2 Associated Arrays `BLANKED` and `LINE`. They are arrays of 0 or 1, set according to the rules exposed in Table 6. More information about these arrays is available in the dedicated **CLASS** documentation.

Table 6: How the Herschel-HIFI FITS flags are translated into the corresponding **CLASS** Associated Arrays. Some flags apply to the whole spectrum, others per channel (marked *ichan*).

CLASS name	Value
`BLANKED[ichan]`	1 if bit 20 (IGNORE_DATA) is set in metacard `rowflag` (level 2.5), else 1 if bit 20 (IGNORE_DATA) is set in column `rowflag` (level 2.0), else 1 if bit 7 (SPUR_CANDIDATE) is set in column `flag[ichan]` (level 2.5), else 1 if bit 7 (SPUR_CANDIDATE) is set in column `flag_*[ichan]` (level 2.0), else 1 if bit 30 (IGNORE_DATA) is set in column `flag[ichan]` (level 2.5), else 1 if bit 30 (IGNORE_DATA) is set in column `flag_*[ichan]` (level 2.0), 0 otherwise.
`LINE[ichan]`	1 if bit 28 (LINE) is set in column `flag[ichan]` (level 2.5), else 1 if bit 28 (LINE) is set in column `flag_*[ichan]` (level 2.0), else 1 if bit 29 (BRIGHT_LINE) is set in column `flag[ichan]` (level 2.5), else 1 if bit 29 (BRIGHT_LINE) is set in column `flag_*[ichan]` (level 2.0), 0 otherwise.

[3] See HIPE documentation for details.

Table 7: Look-up table between the **CLASS** Herschel-HIFI section and the Herschel-HIFI FITS. Depending on the FITS structure (pipeline level), some elements found in a FITS header or in a FITS binary table. FITS names are case sensitive.

CLASS	Type	Value	Default	Description
obsid	I*8	Card OBS_ID	0	Observation id
instrument	C*8	Card INSTRUME	Blank	Instrument name
proposal	C*24	Card PROPOSAL	Blank	Proposal name
aor	C*68	Card AOR	Blank	Astronomical Observation Request
operday	I*4	Card ODNUMBER	0	Operational day number
dateobs	C*28	Card DATE-OBS	Blank	Start date
dateend	C*28	Card DATE-END	Blank	End date
obsmode	C*40	Card OBS_MODE	Blank	Observing mode
vinfo	R*4	Metacard vlsr if card REDSHFT is "optical" or "radio"	0.0	Informative source velocity in LSR
zinfo	R*4	Metacard vlsr if card REDSHFT is not "optical" or "radio", else 0.0	0.0	Informative target redshift
posangle	R*8	Card POSANGLE	0.d0	Spacecraft pointing position angle
reflam	R*8	Card RAOFF	0.d0	Sky reference (a.k.a. OFF) lambda
refbet	R*8	Card DECOFF	0.d0	Sky reference (a.k.a. OFF) beta
hifavelam	R*8	Metacard longitude_cmd (level 2.5) or Column longitude_cmd (level 2.0)	0.d0	HIFI ON average H and V lambda
hifavebet	R*8	Metacard latitude_cmd (level 2.5) or Column latitude_cmd (level 2.0)	0.d0	HIFI ON average H and V beta
etamb	R*4	Card ETAMB	0.0	Main beam efficiency used when applying doAntennaTemp
etal	R*4	Card ETAL	0.0	Forward efficiency used when applying doAntennaTemp
etaa	R*4	Card ETAA	0.0	Telescope aperture efficiency
hpbw	R*4	Card HPBW	0.0	Azimuthally-averaged HPBW
tempscal	C*8	Card TEMPSCAL	Blank	Temperature scale in use
lodopave	R*8	Card LODOPPAV	lofreq (Table 2)	Average LO frequency Doppler corrected to freqFrame (SPECSYS)
gim0	R*4	Metacard *sbGain_0 (usb or lsb)	0.0	Sideband gain polynomial coeff 0
gim1	R*4	Metacard *sbGain_1 (usb or lsb)	0.0	Sideband gain polynomial coeff 1
gim2	R*4	Metacard *sbGain_2 (usb or lsb)	0.0	Sideband gain polynomial coeff 2
gim3	R*4	Metacard *sbGain_3 (usb or lsb)	0.0	Sideband gain polynomial coeff 3
mixercurh	R*4	Metacard MJC_Hor (level 2.5) or Column MJC_Hor (level 2.0)	0.0	Calibrated mixer junction current (horizontal polarization)
mixercurv	R*4	Metacard MJC_Ver (level 2.5) or Column MJC_Ver (level 2.0)	0.0	Calibrated mixer junction current (vertical polarization)
datehcss	C*28	Card DATE	Blank	Processing date
hcssver	C*24	Card CREATOR	Blank	HCSS version
calver	C*16	Card CALVERS	Blank	Calibration version
level	R*4	Card LEVEL	0.	Pipeline level

3 Technical details

3.1 Detecting an Herschel-HIFI FITS

A FITS file fed as input of the command `LAS\FITS` is assumed to be an Herschel-HIFI FITS if the Primary HDU:

1. contains the card `HCSS____`, and
2. the value of the card `TYPE` is not `HICLASS` nor `'Class formatted fits file'`.

The 2nd rule discriminates Herschel-HIFI FITS and **CLASS** FITS produced by `HICLASS`.

3.2 FITS level

The command `LAS\FITS` traverses one by one all the HDUs in the FITS file *independently of their names*. If a binary table is found and:

- if it contains a column `frequency` or `wave`, it is assumed to be a level 2.5 Herschel-HIFI FITS and the specific decoding is started.
- if if contains at least one column `lsbfrequency_*` or `usbfrequency_*`, it is assumed be a level 2.0 Herschel-HIFI FITS and the specific decoding is started.

The levels 2.0 and 2.5 refer to the reduction pipeline level in HCSS. The resulting FITS provides (almost) the same data but under a different form. For our needs, the main differences come from values which are found in columns of the binary table in level 2.0, and in metacards (see below) of the header in level 2.5.

3.3 Cards and metacards

In this document, we refer to 2 concepts in the FITS context:

- header *cards*. In the FITS standard, an header card has the form:

  ```
  CALVERS = 'HIFI_CAL_22_0'      / HIFI calibration version
  ```

  with an 8-characters key, a value, and a comment. In this document, we abusively refer to *e.g.* "the card `CALVERS`" as "the value in the card which key is `CALVERS`".

- header *metacards*. Not in the FITS standard, we refer as metacards when encoutering in the FITS header a pair of lines (contiguous or not, ordered or not) with the form:

  ```
  META_10 =    62.39999999999999 / [s]
  HIERARCH  key.META_10='integrationTime'
  ```

  i.e. a standard card and a `HIERARCH` (free-form) line. We can see that these 2 lines provide an indirection through a `META_XX` key (`XX` being a non-ambiguous number). In this case, we abusively refer to *e.g.* "the metacard `integrationTime`" as "the value in the card which key (`META_10`) has the value `integrationTime` in the associated `HIERARCH` line". Thanks to the metacards, case-sensitive keywords not limited to 8 characters can be used. There is also no need to refer to the exact `META_XX` name in the subsequent specifications, *i.e.* `XX` can vary from one file to another.

4 Acknowledgements

This work was partly founded by CNES. We thanks Claudia Comito and Jonathan Braine for useful comments during this work.